ProQuest®

Return to the USPTO NPL Page | Help

Basic Search Advanced Search Topic Guide Publication Search Marked List : 0 articles Interface language: English

Databases selected: Multiple databases... What's New

Results – powered by ProQuest® Smart Search

Suggested Topics About < Previous | Next >

Shareholders wealth

Shareholders wealth AND Stock prices

Shareholders wealth AND Rates of return

Browse Suggested Publications About < Previous | Next >

Harvard Business Review; Boston

13 articles found for: *(alain rappaport) AND PDN(<6/18/1999)* — all considered keywords/title

All sources Trade Publications Newspapers

□ Mark / Clear all on page | View marked articles Show all documents Sort results by: Most recent articles first

□ 1. **The innovative applications of artificial intelligence conference**
Bruce Buchanan, Sam Uthurusamy. **AI Magazine.** La Canada: Spring 1999. Vol. 20, Iss. 1; p. 11 (2 pages)
Full text Page Image - PDF Citation

□ 2. **Neuron Data names Michael Braun president and CEO**
Smith, Scott. **Business Wire.** New York: Jul 31, 1996. p. 1 (41 pages)
Full text Abstract

□ 3. **Neuron Data lets users build Web links into existing apps**
Cox, John. **Network World.** Framingham: Dec 18, 1995. Vol. 12, Iss. 51; p. 8 (1 page)
Full text Page Image - PDF Abstract

□ 4. **DB tools in ferment**
Vaughan, Jack. **Software Magazine.** Englewood: Oct 1993. Vol. 13, Iss. 15; p. 34 (2 pages)
Full text Page Image - PDF Abstract

□ 5. **Neuron data broadens tool base**
Ballou, Melinda-Carol. **Computerworld.** Framingham: Jul 19, 1993. Vol. 27, Iss. 29; p. 79 (1 page)
Full text Page Image - PDF Abstract

□ 6. **Neuron Data appoints new VP of sales**
Marcroft, Theresa. **Business Wire.** New York: Jan 18, 1993. p. 1
Full text Abstract

□ 7. **Code Trapped Between Legacy, Object Worlds**
Keyes, Jessica. **Software Magazine.** Englewood: Jun 1992. Vol. 12, Iss. 8; p. 39 (5 pages)
Full text Page Image - PDF Abstract

□ 8. **GUI Builders Pay Price for User Productivity**
Bochenski, Barbara. **Software Magazine.** Englewood: Apr 1992. Vol. 12, Iss. 5; p. 74 (5 pages)
Full text Page Image - PDF Abstract

□ 9. **Moving Smartly New Directions in Artificial Intelligence Mean Opportunities for Business; [Orange County Edition]**
DEAN TAKAHASHI. **Los Angeles Times (pre-1997 Fulltext).** Los Angeles, Calif.: Jul 17, 1991. p. 5
Full text Abstract

□ 10. **Books: Innovative Applications in Artificial Intelligence**
Anonymous. **Mechanical Engineering.** New York: Mar 1990. Vol. 112, Iss. 3; p. 90 (1 page)
Page Image - PDF Abstract

□ 11. **Artificial intelligence is starting to get smart; [ME2 Edition]**
Steve Kaufman SPECIAL TO THE STAR. **Toronto Star.** Toronto, Ont.: Jan 8, 1990. p. B.4
Full text Abstract

□ 12. **Palo Alto Firms Apply Reason to Computers; [FINAL Edition]**
Don Clark, Chronicle Staff Writer. **San Francisco Chronicle (pre-1997 Fulltext).** San Francisco, Calif.: Dec 13, 1989. p. C.1
Full text Abstract

□ 13. **Artificial Intelligence: There's Reason Beyond Rules**
Spang, Sara. **Computerworld.** Framingham: Oct 17, 1988. Vol. 22, Iss. 42; p. 88 (1 page)
Page Image - PDF Abstract

1-13 of 13

Did you find what you're looking for? If not, revise your search below or try these suggestions:

Results per page: 30

Suggested Topics About < Previous | Next >

Shareholders wealth
Shareholders wealth AND Stock prices
Shareholders wealth AND Rates of return

Browse Suggested Publications About < Previous | Next >

Harvard Business Review; Boston

Basic Search

Tools: Search Tips Browse Topics 11 Recent Searches

alain rappaport [Search] [Clear]

Database: Multiple databases... Select multiple databases

Date range: Before this date... 06/18/1999 About

Limit results to: ☑ Full text articles only

□ Scholarly journals, including peer-reviewed About

More Search Options

Text-only interface

From:ProQuest

« Back to Article View

Databases selected: Multiple databases... What's New

Code Trapped Between Legacy, Object Worlds

Keyes, Jessica. **Software Magazine.** Englewood: Jun 1992.Vol. 12, Iss. 8; pg. 39, 5 pgs

Subjects:	Technological planning, Technological change, Object oriented programming, Information systems, Computer languages
Classification Codes	9190, 5240, 2310
Locations:	US
Author(s):	Keyes, Jessica
Publication title:	Software Magazine. Englewood: Jun 1992. Vol. 12, Iss. 8; pg. 39, 5 pgs
Source Type:	Periodical
ISSN/ISBN:	08978085
ProQuest document ID:	808554
Text Word Count	3108
Article URL:	http://gateway.proquest.com/openurl?url_ver=Z39.88-2004&res_dat=xri:pqd&rft_val_fmt=info:ofi/fmt:kev:mtx:journal&genre=article&rft_dat=xri:pqd:did=000000000808554&svc_dat=xri:pqil:fmt=text&req_dat=xri:pqil:pq_clntid=19649

Abstract (Article Summary)

While the software trend for many firms is moving toward programming languages that provide reusability benefits, these object-oriented languages, such as C++ and Smalltalk, apply to the workstations, not the mainframe legacy systems that most need them. Some industry observers do not believe that simply moving from Cobol to object-oriented languages will lead to a solution. A business must concentrate first on its goals and strategies, and then translate those goals and strategies into business rules regarding its information systems (IS). The organizations that will be successful in the 1990s are those that develop the capability to change their businesses very rapidly and ensure that their systems can change at the same rate. Since the tools for moving legacy code automatically into object-oriented environment are not available, an interim solution is to reengineer legacy code into an information engineering model. When deciding on a migration to object orientation, IS managers should consider return on investment.

Full Text (3108 words)

Elizabeth Ash and Richard Cullom had a problem. The information systems department develops systems for the 24,000 workers at First Union National Bank. Object-oriented development seemed like a sure route to enhanced productivity. But then there was all that legacy Cobol code.

With an organization our size, Cobol is everywhere and rewriting any of these application programs is a major process," said Ash, assistant vice president and systems manager at the Charlotte, N.C.-based bank holding company. Cullom, a systems developer at First Union, faces the challenge of, as Ash puts it, "using object orientation to accentuate our existing systems."

Cullom's immediate goal is to extend the life of First Union's legacy systems by front-ending these systems with fancy object-oriented and graphical interfaces. First Union's problem is one that many IS shops face. "We can't solve the legacy code problem right now. And while we're waiting, all we can do is provide the data in a format in which folks can use it," he explained.

Cullom said the 350-person IS department works with two distinct technologies. "On the mainframe, we are working to bring in Case (computer-aided software engineering) technology to reorganize and go forward. And on the PC, we are very much into object-oriented programming."

For now Ash and Cullom are "waiting for the fallout in the industry to determine the best way to approach our Cobol systems on the mainframe," said Cullom.

Today, most firms find that they are trapped between two different worlds. Momentum is now moving toward programming languages that provide reusability benefits. Unfortunately, these object-oriented languages, such as C++ and Smalltalk, sit squarely in the center of the workstation world--far from the mainframe legacy systems that need them the most.

Companies such as First Union can only "wrap" the flavor of object orientation around tired and worn legacy systems by providing graphical user interface (GUI) front ends. Like prisoners who can see but cannot reach what is beyond their bars, these companies must wait for the key that unlocks the door to this other world.

This leads to the question of when mainframe legacy systems will be able to take advantage of object-oriented technology.

Some industry observers take exception to the notion that simply moving from one technology (Cobol) to another (object-oriented languages) will, by itself, solve the problem. These observers include Clive Finkelstein and James Martin, developers of information engineering methodologies. Others warn that concentrating on technology at the expense of methodology is dangerous.

Finkelstein and Martin make a strong case for a business to concentrate first on its goals and strategies, and then to translate those goals and strategies into business rules. Only then should a business begin to select a technology to automate those rules.

Finkelstein is the founder of Information Engineering Systems Corp. (IESC), Alexandria, Va. He has been writing, lecturing and practicing information engineering since he and James Martin co-authored the Savant Institute report entitled, "Information Engineering," in 1981. In a new book, entitled Information Engineering: Strategic Systems Development, which Addison-Wesley will publish later this year, Finkelstein describes a new direction for information engineering.

Information engineering, according to Finkelstein, is a key component in the legacy system equation. "Organizations are in constant flux today. They need to be able to change rapidly to survive. Many of the existing systems have not been designed for this rapid change," Finkelstein explained. "To move to object orientation is dangerous."

According to Finkelstein, legacy code contains the business rules of the organization at a particular time. Simply to convert this code is to lose sight of the fact that the business organization is in a state of flux, and as the business changes its rules should be changing accordingly. Therefore, he recommends that the move to object orientation, "should only be done in conjunction with what you need for the future."

The key point is to understand what the organization needs to achieve, said Finkelstein. "You can then, knowing the goals and objectives, derive appropriate strategies. This business point of view leads to establishment of functions and processes, which ultimately lead to systems," he added.

Since change and flux are constants, Finkelstein warns that the only organizations that will be successful in the '90s are those that develop the capability to change their businesses very rapidly. At the same time, they must ensure that their systems can change at the same rate.

Information engineering espouses that the key to good systems is in being able to create a business model. The technology that automates this model is theoretically secondary and interchangeable, and therefore object tools should be able to automate the information engineering model as well as their predecessors.

"The organization can capture that level of knowledge in the business model and then translate it into cohesive logic 'Lego' building blocks, which can be clicked together in every different way and which can change each time it (the model) is used," Finkelstein explained.

Glen Hughlette, president of IESC, agrees. "Having those higher-level representations means that systems can be maintained at a higher level, and ultimately be more flexible and less expensive. Exponential improvements will come from tying business systems into business strategies, which develops a business architecture that everything else can be built on." Hughlette stresses that the organization should focus first on what it needs in the future. Then, he said, the organization should think through the various alternatives before choosing a solution.

Like many industry observers, Martin chairman of James Martin Associates, Chicago, recognizes the immense investment of intellectual capital tied up in legacy systems.

Martin agrees that an organization would be much better off if its code were in an object-oriented form, since it permits better redesign. However, he warns that getting from legacy code to object-oriented code is not an easy process. One problem, according to Martin, is that there are virtually no available tools to help make this great leap.

"We're just seeing the beginning of tools such as Easel (from Easel Corp., Burlington, Mass.) and Mozart (from Mozart Systems, Burlingame, Calif.), which enable you to shrink-wrap a legacy application, such as a mainframe application, and then build a GUI interface to it. Then you can add things outside of the legacy system, such as building add-on modules directly on the PC," he said.

READYING FOR CHANGE

Since the tools are not quite there, how does one get an organization ready for change? "Most organizations are using models today that adhere to conventional structured techniques, with conventional data administration and normalization of data," said Martin. "More advanced organizations are modeling their enterprise in terms of objects. By the year 2000, the advanced corporation is going to have a very thorough object-oriented model.

The practical question, then, is when and how fast should organizations move? "It's practical today to do object-oriented modeling in the enterprise, but the tools aren't here to move legacy code automatically into the object-oriented environment," he said. The interim answer, from Martin's point of view, is to reengineer legacy code into an information engineering model.

Martin claims an organization has seven options in tackling the migration from legacy to modern systems. As an organization moves to a higher-level option, he said, the complexity and expense grow proportionately

In option one, the company finds itself with legacy code that works. Martin's suggestions is to "do nothing.

If it works don't fix it." On the next level is legacy code that requires some minor changes. In this instance, it is more cost-effective to make these changes in the system's present form than to reengineer it. Martin's third option is to restructure the code, but not to reengineer it. An example, he said, is to take "spaghetti Cobol and put it through an engine such as Recoder from Language Technology (Salem, Mass.) and turn it into structured code."

Shrink-wrapping the legacy code is the fourth option. It is perhaps the most popular choice today among IS managers, First Union included. Here the developer freezes the legacy code (unchanged), and makes additions on the outside. According to Martin, this is where object-oriented tools will play a role. "We're going to see object-oriented tools taking the approach of shrink-wrapping legacy code, making it look rather like an object so that you can send object-oriented requests to it."

Martin's fifth option is to reengineer the code. A developer reverse engineers the code into a Case repository so it can ultimately be forward engineered in Case fashion. The key for the future, said Martin, "is to link the reverse engineering to the forward engineering."

Reengineering to an information engineering model is Martin's sixth option. Of course, this implies that the organization possesses data models with the data correctly normalized. "In old systems data is not properly normalized; it's probably a mess. And so here we're saying reverse engineer it into an environment and change the data structures. Even though this is much more complex, and more expensive, it's better in the long run," he said.

The option that carries the highest cost, Martin's seventh option, is to "scrap it and redevelop it." This option is not as expensive as it may appear, however. "Today you're likely to regenerate it with code generators and Case tools, and so the cost of scrapping it is much lower than it would have been with ordinary Cobol," he said. "In ordinary Cobol, the rule of thumb is if more than 12% of the lines of code change, it's cheaper to scrap it and redevelop it than attempt to maintain it."

Like IESC's Finkelstein, Martin thinks that an organization would be making a mistake to force legacy code to object orientation today. "The tools are just not here. The organization would be spending a lot of money and not getting a lot of functionality" he said.

Martin recommends that the IS department should make decisions like a CEO when judging the migration to object orientation. Return on investment, he said, should be the deciding factor in the decision. "Right now this activity would be very expensive, so the ROI would almost certainly be negative." Martin does see some solutions down the road, though. "Eventually, we are going to get very nice object-oriented tools," he said.

First Union National Bank enlisted the help of Knowledge Systems, Cary, N.C., in its move to object orientation. The bank utilized Knowledge Systems' expertise primarily for training in the Smalltalk object-oriented programming language.

Knowledge Systems' goal, according to President Reed Phillips, "is to fit object technology into the environment companies now have. Objects are about what business needs to do." He believes that most organizations recognize that objects are part of the solution.

According to Jay Casler, a senior member of Knowledge Systems' technical staff, "screen scraping," the phrase he uses to describe building a GUI front-end to a legacy system, is only good for a limited set of applications. Tools that provide this sort of functionality are not object-oriented, according to Casler, but more like scripting languages. "These scripting languages simply do not have the expressive power to describe the business and the business rules associated with it," he added.

When Phillips and Casler take on a client, they first determine whether the application is appropriate for object orientation. "Usually the way we can tell is that the application they are using is running out of gas," said Casler. "For instance, we were working with a client that had credit card files sitting in DB2 on the mainframe and image files on an Image Plus system. They needed to be able to put this all together, so they spent a year writing code which never did work. After three or four months they had a working system deployed using object technology."

Knowledge Systems will be marketing a solution this spring that fits into Martin's fourth migration option--shrink-wrapping the legacy code. Labeled internally as Harmony, the solution is described as a client/server computing environment. It is made up of several layers. The first-level user interface is a combination of Smalltalk and a component the company has named a window-builder technology. This component permits the user to construct a window using point-and shoot techniques.

Level two, which Knowledge Systems refers to as team development, permits the organization to access external data, from SQLServer for example, and have that data treated as objects within the Smalltalk environment. A suite of databases is accessible, including mainframe DB2.

At the lowest level of the architecture is the communications area.

"We dealt with complexities of low-level access so that on a higher level people would have the freedom to deal only with objects," said Phillips. The product promises to be a harbinger of things to come.

One company with roots in AI that may disagree with Martin's view of when solutions will be available to bring legacy systems into the object world is Neuron Data, Palo Alto, Calif. The company originally positioned itself in 1986 as a supplier of the expert system development tool, Nexpert Object. In response to demands from its customers, the company began marketing a portable GUI builder tool developed internally. Its Open Interface tool is now responsible for about 50% of the firm's revenues, which were about $15 million in its last fiscal year, according to Patrick Perez, CEO and chairman.

Part of Neuron Data's marketing strategy, is to target software suppliers whose development resources are strained in an effort to support multiple GUIs. One such client is American Management Systems (AMS), Arlington, Va., which sells professional services and proprietary application packages. According to Perez, AMS is using Neuron's Open Interface to provide Microsoft Windows and IBM OS/2 GUI support to its range of mainframe applications. AMS is building a communications application program interface (API) layer that lets Open Interface appear as a 3270 session to the host. Neuron Data provides the API for the GUI client.

"The two worlds are glued together," said Perez.

Once in the door with Open Interface, Neuron Data is in a position to tout the capabilities of Nexpert Object. "This is a truly different technology," said Alain Rappaport, Neuron Data president. "Nexpert is more of a software development tool than a niche AI tool. We have many customers not at all familiar with AI. But once we solve the GUI problem, people want to see how the inference engine can be used to write events related to rules and objects."

CONVERTING THE COBOL WORLD

Jerome Garfunkel, president of Jerome Garfunkel Associates in Manhasset, N.Y, is an authority on Cobol. He sees the Cobol languages as sitting in the middle of this picture. "There's no way that object-oriented technology will survive unless the Cobol world converts over," he said.

Like industry guru Martin, Garfunkel recognizes the huge investment in intellectual capital tied up in old Cobol code. Gargunkel is a member of the Object-Oriented Cobol Task Group, which is working hard to make Cobol object-oriented. Garfunkel believes that it is a natural transition. "The main theory behind all of the work that Yourdon, Constantine and DeMarco did, is that if you break code up you reduce the overall complexity. That's the basis of all structured programming. The same premise is absolutely true for object orientation."

Marc Sokol is also a member of the Object-Oriented Cobol Task Group, as well as product owner of the Realia product line of PC Cobol compilers now owned by Computer Associates International (CA), Inc., Islandia, N.Y. "The language is the least of the issue. What is important is the development and maintenance tools that go along with it," Sokol said.

According to CA research, "People are not chomping at the bit to have an object-oriented Cobol." With that in mind, CA is looking a bit longer down the road and "really thinking things through before we announce anything," said Sokol.

Micro Focus, Inc., Palo Alto, Calif., has not made an official announcement, but is in the midst of developing object-oriented extensions to a version of its Cobol.

Dan Clarke, product manager for object-oriented products, sums up the Cobol and object-orientation conundrum. "There's a Band-Aid answer and a long-term answer. The Band-Aid is the wrapper. What that means is to take legacy code and put a wrapper around it so the code can participate in object-oriented systems. The long-term solution is to add object-oriented features to Cobol itself," he said.

Clarke does recognize that even though vendors may add object-oriented features to Cobol, organizations will still need to communicate with legacy Cobol. "We will use these wrappers around the legacy Cobol code so that it understands these messages and responds. Actually, the wrapper is more of a technique than a product," he added.

Micro Focus' evaluation program is a set of language extensions wrapped up in a workbench. "The workbench concept contains a Reusable Code Manager which permits the developer to define macros and provides an object-oriented pre-processor. The workbench will also enable developers to scan through old Cobol programs and break them into modules. This is done through the Cobol Source Information Module," explained Clarke.

The Cobol Source Information Module is a precursor to being able to navigate automatically through legacy Cobol code and turn it into objects. It enables the developer to scan legacy code and obtain a cross-reference listing, along with a set of suggestions as to next steps. Micro Focus is working with Mountain View, Calif.-based ParcPlace Systems, a leading vendor of Smalltalk products, to take this to the next step--deconstruction of a Cobol program into object-oriented code.

Micro Focus is also working with Digitalk, Los Angeles, which has licensed Micro Focus to distribute its version of Smalltalk. "We want to be able to nibble off a piece of the Cobol application and do it in Smalltalk. What we've done is provide extensions to the class hierarchies that come with Smalltalk," said Micro Focus' Clarke. "These extensions bind to our common Communication Interface Module that is part of the workbench, so we have a communication switch which enables you to use Smalltalk as the client and Cobol as the server."

For now, vendors like Micro Focus and Knowledge Systems are leading the Cobol charge. But companies like First Union Bank are waiting for the Case companies to take up the cause. And players with different roots such as Neuron Data, ParcPlace and Digitalk are also in the running.

First Union's Cullom sums up the way most IS professionals feel: "We needed some solutions in a hurry. That's why we turned to Smalltalk to front-end our systems. In the future, the way we move depends on how successful we are, the fall out in the industry, and the direction people take in major industries according to what object orientation can provide."